

Prisoners of their own device: Trojan attacks on device-independent quantum cryptography

Jonathan Barrett,^{1,*} Roger Colbeck,^{2,3,†} and Adrian Kent^{4,3,‡}

¹*Department of Mathematics, Royal Holloway, University of London, Egham Hill, Egham, TW20 0EX, U.K.*

²*Institute for Theoretical Physics, ETH Zurich, 8093 Zurich, Switzerland.*

³*Perimeter Institute for Theoretical Physics, 31 Caroline Street North, Waterloo, ON N2L 2Y5, Canada.*

⁴*Centre for Quantum Information and Foundations, DAMTP, Centre for Mathematical Sciences, University of Cambridge, Wilberforce Road, Cambridge, CB3 0WA, U.K.*

(Dated: 11th October 2012)

Device-independent quantum cryptographic schemes aim to guarantee security to users based only on the output statistics of any components used, and without the need to verify their internal functionality. Since this would protect users against untrustworthy or incompetent manufacturers, sabotage or device degradation, this idea has excited much interest, and many device-independent schemes have been proposed. Here we identify a critical weakness of device-independent protocols that rely on public communication between secure laboratories. Untrusted devices may record their inputs and outputs and reveal information about them via publicly discussed outputs during later runs. Reusing devices thus compromises the security of a protocol and risks leaking secret data. Possible defences include securely destroying or isolating used devices. However, these are costly and often impractical. We propose other more practical partial defences as well as a new protocol structure for device-independent quantum key distribution that aims to achieve composable security in the case of two parties using a small number of devices to repeatedly share keys with each other (and no other party).

Quantum cryptography aims to exploit the properties of quantum systems to ensure the security of various tasks. The best known example is quantum key distribution (QKD), which can enable two parties to share a secret random string and thus exchange messages secure against eavesdropping, and we mostly focus on this task for concreteness. While all classical key distribution protocols rely for their security on assumed limitations on an eavesdropper's computational power, the advantage of quantum key distribution protocols (e.g. [1, 2]) is that they are provably secure against an arbitrarily powerful eavesdropper, even in the presence of realistic levels of losses and errors [3]. However, the security proofs require that quantum devices function according to particular specifications. Any deviation – which might arise from a malicious or incompetent manufacturer, or through sabotage or degradation – can introduce exploitable security flaws (see e.g. [4] for practical illustrations).

The possibility of quantum devices with deliberately concealed flaws, introduced by an untrustworthy manufacturer or saboteur, is particularly concerning, since (i) it is easy to design quantum devices that appear to be following a secure protocol but are actually completely insecure¹, and (ii) there is no general technique for identifying all possible security loopholes in standard quantum cryptography devices. This has led to much interest

in device-independent quantum protocols, which aim to guarantee security *on the fly* by testing the device outputs [5–15]: no specification of their internal functionality is required.

Known provably secure schemes for device-independent quantum key distribution are inefficient, as they require either independent isolated devices for each entangled pair to ensure device-independent security [6, 10–12, 16], or a large number of entangled pairs to generate a short key [6, 16, 17]. Finding an efficient secure device-independent quantum key distribution scheme using two (or few) devices has remained an open theoretical challenge. Nonetheless, in the absence of tight theoretical bounds on the scope for device-independent quantum cryptography, progress to date has encouraged optimism (e.g. [18]) about the prospects for device-independent QKD as a practical technology, as well as for device-independent quantum randomness expansion [13–15] and other applications of device-independent quantum cryptography (e.g. [19]).

However, one key question has been generally neglected in work to date on device-independent quantum cryptography, namely what happens if and when devices are reused. Specifically, are device-reusing protocols *composable* – i.e. do individually secure protocols of this type remain secure when combined? It is clear that reuse of untrusted devices cannot be *universally composable*, i.e. such devices cannot be securely reused for completely general purposes (in particular, if they have memory, they must be kept secure after the protocol). However, for device-independent quantum cryptography to have significant practical value, one would hope that devices can at least be reused for the same purpose. For example one would like to be able to implement a QKD

*Electronic address: jon.barrett@rhul.ac.uk

†Electronic address: colbeck@phys.ethz.ch

‡Electronic address: a.p.a.kent@damtp.cam.ac.uk

¹ In BB84 [1], for example, a malicious state creation device could be programmed to secretly send the basis used for the encoding in an additional degree of freedom.

protocol many times, perhaps with different parties each time, with a guarantee that all the generated keys can be securely used in an arbitrary environment so long as the devices are kept secure. We focus on this type of composability here.

We describe a new type of attack that highlights pitfalls in producing protocols that are composable (in the above sense) with device-independent security for reusable devices, and show that for all known protocols such composability fails in the strong sense that purportedly secret data become completely insecure. The leaks do not exploit new side channels (which proficient users are assumed to block), but instead occur through the device choosing its outputs as part of a later protocol.

To illustrate this, consider a device-independent scheme that allows two users (Alice and Bob) to generate and share a purportedly secure cryptographic key. A malicious manufacturer (Eve) can design devices so that they record and store all their inputs and outputs. A well designed device-independent protocol can prevent the devices from leaking information about the generated key *during that protocol*. However, *when they are reused*, the devices can make their outputs in later runs depend on the inputs and outputs of earlier runs, and, if the protocol requires Alice and Bob to publicly exchange at least some information about these later outputs (as all existing protocols do), this can leak information about the original key to Eve. Moreover, in many existing protocols, such leaks can be surreptitiously hidden in the noise, hence allowing the devices to operate indefinitely like hidden spies, apparently complying with security tests, and producing only data in the form the protocols require, but nonetheless actually eventually leaking all the purportedly secure data.

We stress that our results certainly do not imply that quantum key distribution *per se* is insecure or impractical. In particular, our attacks do not apply to standard QKD protocols in which the devices' properties are fully trusted, nor if the devices are trusted to be memoryless (but otherwise untrusted), nor necessarily to protocols relying on some other type of partially trusted devices. Our target is the possibility of (full) device-independent quantum cryptographic security, applicable to users who purchase devices from a potentially sophisticated adversarial supplier and rely on no assumption about the devices' internal workings.

The attacks we present raise new issues of composability and point towards the need for new protocol designs. We discuss some countermeasures to our attacks that appear effective in the restricted but relevant scenario where two users only ever use their devices for QKD exchanges with one another, and propose a new type of protocol that aims to achieve security in this scenario while allowing device reuse. Even with these countermeasures, however, we show that security of a key generated with Bob can be compromised if Alice uses the same device for key generation with an additional party. This appears to be a generic problem against which we see no complete

defence.

Although we focus on device-independent QKD for most of this work, our attacks also apply to other device-independent quantum cryptographic tasks. The case of randomness expansion is detailed in Appendix E.

Cryptographic scenario.—We use the standard cryptographic scenario for key distribution between Alice and Bob, each of whom has a secure laboratory. These laboratories may be partitioned into secure sub-laboratories, and we assume Alice and Bob can prevent communication between their sub-laboratories as well as between their labs and the outside world, except as authorized by the protocol. The setup of these laboratories is as follows. Each party has a trusted private random string, a trusted classical computer and access to two channels connecting them. The first channel is an insecure quantum channel. Any data sent down this can be intercepted and modified by Eve, who is assumed to know the protocol. The second is an authenticated classical channel which Eve can listen to but cannot impersonate; in efficient QKD protocols this is typically implemented by using some key bits to authenticate communications over a public channel. Each party also uses a sub-laboratory to isolate each of the untrusted devices being used for today's protocol. They can connect them to the insecure quantum channel, as desired, and this connection can be closed thereafter. They can also interact with each device classically, supplying inputs (chosen using the trusted private string) and receiving outputs, without any other information flowing into or out of the secure sub-laboratory.

As mentioned before, existing device-independent QKD protocols that have been proven unconditionally secure [6, 11, 12] require separate devices for each measurement performed by Alice and Bob with no possibility of signalling between these devices², or are inefficient [17] (in terms of the amount of key per entangled pair). For practical device-independent QKD, we would like to remove both of these disadvantages and have an efficient scheme needing a small number of devices.

Since the protocols in [11, 12] can tolerate reasonable levels of noise and are reasonably efficient, we look first at implementations of protocols taking the form of those in [11, 12], except that Alice and Bob use one measurement device each, i.e., Alice (Bob) uses the same device to perform each of her (his) measurements. We call these *two-device* protocols (Bob also has a separate isolated source device: see below). The memory of a device can then act as a signal from earlier to later measurements, hence the security proofs of [11, 12] do not apply (see also [20] where a different two-device setup is discussed). It is an open question whether a secure key can be efficiently generated by a protocol of this type in this

² Within the scenario described above, this could be achieved by placing each device in its own sub-laboratory.

1. Entangled quantum states used in the protocol are generated by a device Bob holds (which is separate and kept isolated from his measurement device) and then shared over an insecure quantum channel with Alice's device. Bob feeds his half of each state to his measurement device. Once the states are received, the quantum channel is closed.
2. Alice and Bob each pick a random input A_i and B_i to their device, ensuring they receive an output bit (X_i and Y_i respectively) before making the next input (so that the i -th output cannot depend on future inputs). They repeat this M times.
3. Either Alice or Bob (or both) publicly announces their measurement choices, and the relevant party checks that they had a sufficient number of suitable input combinations for the protocol. If not, they abort.
4. (*Sifting.*) Some output pairs may be discarded according to some public protocol.
5. (*Parameter estimation.*) Alice randomly and independently decides whether to announce each remaining bit to Bob, doing so with probability μ (where $M\mu \gg 1$). Bob uses the communicated bits and his corresponding outputs to compute some test function, and aborts if it lies outside a desired range. (For example, Bob might compute the CHSH value [21] of the announced data, and abort if it is below 2.5.)
6. (*Error correction.*) Alice and Bob perform error correction using public discussion, in order to (with high probability) generate identical strings. Eve learns the error correction function Alice applies to her string.
7. (*Privacy amplification.*) Alice and Bob publicly perform privacy amplification [22], producing a shorter shared string about which Eve has virtually no information. Eve similarly learns the privacy amplification function they apply to their error-corrected strings.

TABLE I: **Generic structure of the protocols we consider.** Although this structure is potentially restrictive, most protocols to date are of this form (we discuss modifications later). Note that we do not need to specify the precise sub-protocols used for error correction or privacy amplification. For an additional remark, see Part I of the Appendix

scenario. Here we demonstrate that, even if a key can be securely generated, repeat implementations of the protocol using the same devices can render an earlier generated key insecure.

Attacks on two-device protocols.—Consider a QKD protocol with the standard structure shown in Table I. We imagine a scenario in which a protocol of this type is run on day 1, generating a secure key for Alice and Bob, while informing Eve of the functions used by Alice for error correction and privacy amplification (for simplicity we assume the protocol has no sifting procedure (Step 4)). The protocol is then rerun on day 2, to generate a second key, using the same devices. Eve can instruct the devices to proceed as follows. On day 1, they follow the protocol honestly. However, they keep hidden records of all the

raw bits they generate during the protocol. At the end of day 1, Eve knows the error correction and privacy amplification functions used by Alice and Bob to generate the secure key.

On day 2, since Eve has access to the insecure quantum channel over which the new quantum states are distributed, she can surreptitiously modulate these quantum states to carry new classical instructions to the device in Alice's lab, for example using additional degrees of freedom in the states. These instructions tell the device the error correction and privacy amplification functions used on day 1, allowing it to compute the secret key generated on day 1. They also tell the device to deviate from the honest protocol for randomly selected inputs, by producing as outputs specified bits from this secret key. (For example, “for input 17, give day 1's key bit 5 as output”.) If any of these selected outputs are among those announced in Step 5, Eve learns the corresponding bits of day 1's secret key. We call this type of attack, in which Eve attempts to gain information from the classical messages sent in Step 5, a *parameter estimation attack*.

If she follows this cheating strategy for $N\mu^{-1} < M$ input bits, Eve is likely to learn roughly N bits of day 1's secret key. Moreover, only the roughly N output pairs from this set that are publicly compared give Alice and Bob statistical information about Eve's cheating. Alice and Bob cannot a priori identify these cheating output pairs among the $\approx \mu M$ they compare. Thus, if the tolerable noise level is comparable to $N\mu^{-1}M^{-1}$, Eve can (with high probability) mask her cheating as noise. (Note that in unconditional security proofs it is generally assumed that eavesdropping is the cause of all noise. Even if in practice Eve cannot reduce the noise to zero, she can supply less noisy components than she claims and use the extra tolerable noise to cheat).

In addition, Alice and Bob's devices each separately have the power to cause the protocol to abort on any day of their choice. Thus – if she is willing to wait long enough – Eve can program them to communicate some or all information about their day 1 key, for instance by encoding the relevant bits as a binary integer $N = b_1 \dots b_m$ and choosing to abort on day $(N+2)^3$. We call this type of attack an *abort attack*. Note that it cannot be detected until it is too late.

As mentioned above, some well known protocols use many independent and isolated measurement devices. These protocols are also vulnerable to memory attacks, as explained in Appendix D.

Modified protocols.—We now discuss ways in which these

³ In practice, Eve might infer a day $(N+2)$ abort from the fact that Alice and Bob have no secret key available on day $(N+2)$, which in many scenarios might detectably affect their behaviour then or subsequently. Note too that she might alternatively program the devices to abort on every day from $(N+2)$ onwards if this made N more easily inferable in practice.

attacks can be partly defended against.

Countermeasure 1.—All quantum data and all public communication of output data in the protocol come from one party, say Bob. Thus, the entangled states used in the protocol are generated by a separate isolated device held by Bob (as in the protocol in Table 1) and Bob (rather than Alice) sends selected output data over a public channel in Step 5. If Bob’s device is forever kept isolated from incoming communication, Eve has no way of sending it instructions to calculate and leak secret key bits from day 1 (or any later day).

Existing protocols modified in this way are still insecure if reused, however. For example, in a modified parameter estimation attack, Eve can pre-program Bob’s device to leak raw key data from day 1 via output data on subsequent days, at a low enough rate (compared to the background noise level) that this cheating is unlikely to be detected. If the actual noise level is lower than the level tolerated in the protocol, and Eve knows both (a possibility Alice and Bob must allow for), she can thereby eventually obtain all Bob’s raw key data from day 1, and hence the secret key.

In addition, Eve can still communicate with Alice’s device, and Alice needs to be able to make some public communication to Bob, if only to abort the protocol. Eve can thus obtain secret key bits from day 1 on a later day using an abort attack.

Countermeasure 2. [23]—Encrypt the parameter estimation information sent in Step 5 with some initial pre-shared seed randomness. Provided the seed required is small compared to the size of final string generated (which is the case in efficient QKD protocols [11, 12]), the protocol then performs key expansion⁴. Furthermore, even if they have insufficient initial shared key to encrypt the parameter estimation information, Alice and Bob could communicate the parameter estimation information unencrypted on day 1, but encrypt it on subsequent days using generated key.

Note that this countermeasure is not effective against abort attacks, which can now be used to convey all or part of their day 1 raw key. This type of attack seems unavoidable in any standard cryptographic model requiring composability and allowing arbitrarily many device reuses if either Alice or Bob has only a single measurement device.

This countermeasure is also not effective in general cryptographic environments involving communication with multiple users who may not all be trustworthy. Suppose that Alice wants to share key with Bob on day 1, but with Charlie on day 2. If Charlie becomes corrupted by Eve, then, for example by hiding data in the parameter estimation, Eve can learn about day 1’s

key (we call this an *impostor attack*). This attack applies in many scenarios in which users might wish to use device-independent QKD. For example, suppose Alice is a merchant and Bob is a customer who needs to communicate his credit card number to Alice via QKD to complete the sale. The next day, Eve can pose as a customer, carry out her own QKD exchange with Alice, and extract information about Bob’s card number without being detected.

Countermeasure 3.—Alternative protocols using additional measurement devices. Suppose Alice and Bob each have m measurement devices, for some small integer $m \geq 2$. They perform Steps 1–6 of a protocol that takes the form given in Table I but with Countermeasures 1 and 2 applied. They repeat these steps for each of their devices in turn, ensuring no communication between any of them (i.e., they place each in its own sub-laboratory). This yields m error-corrected strings. Alice and Bob concatenate their strings before performing privacy amplification as in Step 7. However, they further shorten the final string such that it would (with near certainty) remain secure if one of the m error-corrected strings were to become known to Eve through an abort attack. (See Table 2, and Appendix C for more details).

This countermeasure doesn’t avoid impostor attacks. Instead, the idea is to prevent useful abort attacks (as well as parameter estimation attacks due to Countermeasure 2), and hence give us a secure and composable protocol, provided the keys produced on successive days are always between the same two users. The information each device has about day 1’s key is limited to the raw key it produced. Thus, if each device is programmed to abort on a particular day that encodes their day 1 raw key, after an abort, Eve knows one of the devices’ raw keys and has some information on the others (since she can exclude certain possibilities based on the lack of abort by those devices so far). After an abort, Alice and Bob should cease to use any of their devices unless and until such time that they no longer require that their keys remain secret. Intuitively, provided the set of m keys was sufficiently shortened in the privacy amplification step, Eve has essentially no information about the day 1 secret key, which thus (we conjecture) remains secure.

Countermeasure 4.—Alice and Bob share a small initial secret key and use part of it to choose the privacy amplification function in Step 7 of the protocol, which may then never become known to Eve.

Even in this case, Eve can pre-program Bob’s measurement device to leak raw data from day 1 on subsequent days, either via a parameter estimation attack or via an abort attack. While Eve cannot obtain bits of the secret key so directly in this case, provided the protocol is composed sufficiently many times, she can eventually obtain all the raw key. This means that Alice and Bob’s residual security ultimately derives only from the initial shared secret key: their QKD protocol produces no extra permanently secure data.

⁴ QKD is often referred to as quantum key expansion in any case, taking into account that a common method of authenticating the classical channel uses pre-shared randomness.

In summary, we have shown how a malicious manufacturer who wishes to mislead users or obtain data from them can equip devices with a memory and use it in programming them. The full scope of this threat seems to have been overlooked in the literature on device-independent quantum cryptography to date. A task is potentially vulnerable to our attacks if it involves secret data generated by devices and if Eve can learn some function of the device outputs in a subsequent protocol. Since even causing a protocol to abort communicates some information to Eve, the class of tasks potentially affected is large indeed. In particular, for one of the most important applications, QKD, none of the protocols so far proposed remain compositably secure in the case that the devices are supplied by a malicious adversary.

One can think of the problems our attacks raise as a new issue of cryptographic composability. One way of thinking of standard composability is that a secure output from a protocol must still have all the properties of an ideal secure output when combined with other outputs from the same or other protocols. The device-independent key distribution protocols we have examined fail this test because the reuse of devices can cause later outputs to depend on earlier ones. In a sense, the underlying problem is that the *usage of devices* is not compositably secure. This applies too, of course, for devices used in different protocols: devices used for secure randomness expansion cannot then securely be used for key distribution without potentially compromising the generated randomness, for example.

It is worth reiterating that our attacks do not apply against protocols where the devices are trusted to be memoryless. Indeed, there are schemes that are compositably secure for memoryless devices [11, 12]. We also stress that our attacks do not apply to all protocols for device-independent quantum tasks related to cryptography. For example, even devices with memories cannot mimic nonlocal correlations in the absence of shared entanglement [24, 25]. In addition, in applications that require only short-lived secrets, devices may be reused once such secrets are no longer required. Partially secure device-independent protocols for bit commitment and coin tossing [19], in which the committer supplies devices to the recipient, are also immune from our attacks, so long as the only data entering the devices come from the committer.

Note too that, in practice the number of uses required to apply the attacks may be very large, for example, in the case of some of the abort attacks we described. One can imagine a scenario in which Alice and Bob want to carry out device-independent QKD no more than n times for some fixed number n , each is confident in the other's trustworthiness throughout, the devices are used for no other purpose and are destroyed after n rounds, and key generation is suspended and the devices destroyed if a single abort occurs. If the only relevant information con-

veyed to Eve is that an abort occurs on one of the n days, she can only learn at most $\log n$ bits of information about the raw key via an abort attack. Hence one idea is that, using suitable additional privacy amplification, Alice and Bob could produce a device-independent protocol using two measurement devices that is provably secure when restricted to no more than n bilateral uses. It would be interesting to analyse this possibility, which, along with the protocol presented in Table 2, leads us to hold out the hope of useful security for fully device-independent QKD, albeit in restricted scenarios.

We have also discussed some possible defences and countermeasures against our attacks. A theoretically simple one is to dispose of – i.e. securely destroy or isolate – untrusted devices after a single use (see Appendix B). While this would restore universal composability, it is clearly costly and would severely limit the practicality of device-independent quantum cryptography. Another interesting possibility is to design protocols for composable device-independent QKD guaranteed secure in more restricted scenarios. However, the impostor attacks described above appear to exclude the possibility of compositably secure device-independent QKD when the devices are used to exchange key with several parties.

Many interesting questions remain open. Nonetheless, the attacks we have described merit a serious reappraisal of current protocol designs and, in our view, of the practical scope of universally composable quantum cryptography using completely untrusted devices.

Added Remark: Since the first version of this paper, there has been new work in this area that, in part, explores countermeasure 2 in more detail [26]. In addition, two new works on device-independent QKD with only two devices have appeared [27, 28]. Note that these do not evade the attacks we present, but apply to the scenario where used devices are discarded.

Acknowledgements.—We thank Anthony Leverrier and Gonzalo de la Torre for [23], Lluís Masanes, Serge Massar and Stefano Pironio for helpful comments. JB was supported by the EPSRC, and the CHIST-ERA DIQIP project. RC acknowledges support from the Swiss National Science Foundation (grants PP00P2-128455 and 20CH21-138799) and the National Centre of Competence in Research ‘Quantum Science and Technology’. AK was partially supported by a Leverhulme Research Fellowship, a grant from the John Templeton Foundation, and the EU Quantum Computer Science project (contract 255961). This research is supported in part by Perimeter Institute for Theoretical Physics. Research at Perimeter Institute is supported by the Government of Canada through Industry Canada and by the Province of Ontario through the Ministry of Research and Innovation.

-
- [1] Bennett, C. H. & Brassard, G. Quantum cryptography: Public key distribution and coin tossing. In *Proceedings of IEEE International Conference on Computers, Systems, and Signal Processing*, 175–179. IEEE (New York, 1984).
- [2] Ekert, A. K. Quantum cryptography based on Bell’s theorem. *Physical Review Letters* **67**, 661–663 (1991).
- [3] Renner, R. *Security of Quantum Key Distribution*. Ph.D. thesis, Swiss Federal Institute of Technology, Zurich (2005). Also available as [quant-ph/0512258](#).
- [4] Gerhardt, I. *et al.* Full-field implementation of a perfect eavesdropper on a quantum cryptography system. *Nature Communications* **2**, 349 (2011).
- [5] Mayers, D. & Yao, A. Quantum cryptography with imperfect apparatus. In *Proceedings of the 39th Annual Symposium on Foundations of Computer Science (FOCS-98)*, 503–509 (IEEE Computer Society, Los Alamitos, CA, USA, 1998).
- [6] Barrett, J., Hardy, L. & Kent, A. No signalling and quantum key distribution. *Physical Review Letters* **95**, 010503 (2005).
- [7] Acín, A., Gisin, N. & Masanes, L. From Bell’s theorem to secure quantum key distribution. *Physical Review Letters* **97**, 120405 (2006).
- [8] Scarani, V. *et al.* Secrecy extraction from no-signaling correlations. *Physical Review A* **74**, 042339 (2006).
- [9] Acín, A. *et al.* Device-independent security of quantum cryptography against collective attacks. *Physical Review Letters* **98**, 230501 (2007).
- [10] Masanes, L., Renner, R., Christandl, M., Winter, A. & Barrett, J. Unconditional security of key distribution from causality constraints. e-print [quant-ph/0606049v4](#) (2009).
- [11] Hänggi, E. & Renner, R. Device-independent quantum key distribution with commuting measurements. e-print [arXiv:1009.1833](#) (2010).
- [12] Masanes, L., Pironio, S. & Acín, A. Secure device-independent quantum key distribution with causally independent measurement devices. *Nature Communications* **2**, 238 (2011).
- [13] Colbeck, R. *Quantum and Relativistic Protocols For Secure Multi-Party Computation*. Ph.D. thesis, University of Cambridge (2007). Also available as [arXiv:0911.3814](#).
- [14] Pironio, S. *et al.* Random numbers certified by Bell’s theorem. *Nature* **464**, 1021–1024 (2010).
- [15] Colbeck, R. & Kent, A. Private randomness expansion with untrusted devices. *Journal of Physics A* **44**, 095305 (2011).
- [16] Barrett, J., Kent, A. & Pironio, S. Maximally non-local and monogamous quantum correlations. *Physical Review Letters* **97**, 170409 (2006).
- [17] Barrett, J., Colbeck, R. & Kent, A. Unconditionally secure device-independent quantum key distribution with only two devices. e-print [arXiv:1209.0435](#) (2012).
- [18] Ekert, A. Less reality, more security. *Physics World* (September 2009).
- [19] Silman, J. *et al.* Fully distrustful quantum bit commitment and coin flipping. *Physical Review Letters* **106**, 220501 (2011).
- [20] Hänggi, E., Renner, R. & Wolf, S. The impossibility of non-signalling privacy amplification. e-print [arXiv:0906.4760](#) (2009).
- [21] Clauser, J. F., Horne, M. A., Shimony, A. & Holt, R. A. Proposed experiment to test local hidden-variable theories. *Physical Review Letters* **23**, 880–884 (1969).
- [22] Bennett, C. H., Brassard, G. & Robert, J.-M. Privacy amplification by public discussion. *SIAM Journal on Computing* **17**, 210–229 (1988).
- [23] de la Torre, G. & Leverrier, A. (2012). Personal communication.
- [24] Barrett, J., Collins, D., Hardy, L., Kent, A. & Popescu, S. Quantum nonlocality, Bell inequalities, and the memory loophole. *Physical Review A* **66**, 042111 (2002).
- [25] Gill, R. D. Accardi contra Bell (cum mundi): The impossible coupling. In Moore, M., Froda, S. & Léger, C. (eds.) *Mathematical Statistics and Applications: Festschrift for Constance van Eeden*, vol. 42 of *IMS Lecture Notes – Monograph Series*, 133–154 (2003).
- [26] McKague, M. & Sheridan, L. Reusing devices with memory in device independent quantum key distribution. e-print [arXiv:1209.4696](#) (2012).
- [27] Reichardt, B. W., Unger, F. & Vazirani, U. Classical command of quantum systems via rigidity of CHSH games. e-print [arXiv:1209.0449](#) (2012).
- [28] Vazirani, U. & Vidick, T. Fully device independent quantum key distribution. e-print [arXiv:1210.1810](#) (2012).
- [29] Carter, J. L. & Wegman, M. N. Universal classes of hash functions. *Journal of Computer and System Sciences* **18**, 143–154 (1979).
- [30] Wegman, M. N. & Carter, J. L. New hash functions and their use in authentication and set equality. *Journal of Computer and System Sciences* **22**, 265–279 (1981).
- [31] Tomamichel, M., Renner, R., Schaffner, C. & Smith, A. Leftover hashing against quantum side information. In *Proceedings of the 2010 IEEE Symposium on Information Theory (ISIT10)*, 2703–2707 (2010).
- [32] Trevisan, L. Extractors and pseudorandom generators. *Journal of the ACM* **48**, 860–879 (2001).
- [33] De, A., Portmann, C., Vidick, T. & Renner, R. Trevisan’s extractor in the presence of quantum side information. e-print [arXiv:0912.5514](#) (2009).
- [34] Tomamichel, M., Colbeck, R. & Renner, R. Duality between smooth min- and max-entropies. *IEEE Transactions on information theory* **56**, 4674–4681 (2010).
- [35] Fehr, S., Gelles, R. & Schaffner, C. Security and composability of randomness expansion from Bell inequalities. e-print [arXiv:1111.6052](#) (2011).
- [36] Vazirani, U. & Vidick, T. Certifiable quantum dice or, testable exponential randomness expansion. e-print [arXiv:1111.6054](#) (2011).
- [37] Pironio, S. & Massar, S. Device-independent randomness expansion secure against quantum adversaries. e-print [arXiv:1111.6056](#) (2011).

Appendix A: Separation of sources and measurement devices

We add here one important comment about the general structure of the generic protocol given in Table 1 of the main text. There it was crucial that in Step 1, in the

case where Bob (rather than Eve) supplies the states, he does so using a device that is isolated from his measurement device. If, on the other hand, Bob had only a single device that both supplies states and performs measurements, then his device can hide information about day 1's raw key in the states he sends on day 2. (This can be done using states of the form specified in the protocol, masking the errors as noise as above. Alternatively, the data could be encoded in the timings of the signals or in quantum degrees of freedom not used in the protocol.)

Appendix B: Toxic device disposal

As noted in the main text, standard cryptographic models postulate that the parties can create secure laboratories, within which all operations are shielded from eavesdropping. Device-independent quantum cryptographic models also necessarily assume that devices within these laboratories cannot signal to the outside – otherwise security is clearly impossible. Multi-device protocols assume that the laboratories can be divided into effectively isolated sub-laboratories, and that devices in separate sub-laboratories cannot communicate. In other words, Alice and Bob must be able to build arbitrary configurations of screening walls, which prevent communication among Eve and any of her devices, and allow only communications specified by Alice and Bob.

Given this, there is no problem *in principle* in defining protocols which prescribe that devices must be permanently isolated: the devices simply need to be left indefinitely in a screened sub-laboratory. While this could be detached from the main working laboratory, it must be protected indefinitely: screening wall material and secure space thus become consumed resources. And indeed in some situations, it may be more efficient to isolate devices, rather than securely destroy them, since devices can be reused once the secrets they know have become public by other means. For example, one may wish to securely communicate the result of an election before announcing it, but once it is public, the devices used for this secure communication could be safely reused.

The alternative, securely destroying devices and then eliminating them from the laboratory, preserves laboratory space but raises new security issues: consider, for example, the problems in disposing of a device programmed to change its chemical composition depending on its output bit.

That said, no doubt there are pretty secure ways of destroying devices, and no doubt devices could be securely isolated for long periods. However, the costs and problems involved, together with the costs of renewing devices, make us query whether these are really viable paths for practical device-independent quantum cryptography.

Appendix C: Privacy Amplification

Here we briefly outline the important features of privacy amplification, which is a key step in the protocol. As explained in the main text, the idea is to compress the string such that (with high probability) an eavesdropper's knowledge is reduced to nearly zero. This usually works as follows. Suppose Alice and Bob share some random string, X , which may be correlated with a quantum system, E , held by the eavesdropper. Alice also holds some private randomness, R . The state held by Alice and Eve then takes the form

$$\rho_{XRE} = \sum_{x,r} P_X(x) P_R(r) |x\rangle\langle x|_X \otimes |r\rangle\langle r|_R \otimes \rho_E^x,$$

where $\{\rho_E^x\}_x$ are normalized density operators, and $P_R(r) = 1/|R|$. The randomness R is used to choose a function $f_R \in \mathcal{F}$, where \mathcal{F} is some suitably chosen set, to apply to X such that, even if she learns R , the eavesdropper's knowledge about the final string is close to zero. If we call the final string $S = f_R(X)$, then Eve has no knowledge about it if the final state takes the form $\tau_S \otimes \rho_{RE}$, where τ_S is maximally mixed on S . However, we cannot usually attain such a state, and instead measure the success of a protocol by its variation from this ideal, measured using the *trace distance*, D . Denoting the final state (after applying the function) by ρ_{SRE} , we are interested in $D(\rho_{SRE}, \tau_S \otimes \rho_{RE})$.

Fortunately, several sets of function are known for which the above distance can be made arbitrarily small. Two common constructions are those based on *two-universal hash functions* [3, 29–31] and *Trevisan's extractor* [32, 33]. The precise details of these is not very important for the present work (we refer the interested reader to the references), nor is it important which we choose. However, it is worth noting that for two-universal hash functions, the size of the seed needs to be roughly equal to that of the final string, while for Trevisan's extractor, this can be reduced to roughly the logarithm of the length of the initial string (in the latter case, this may allow it to be sent privately, if desired).

For both, the amount that the string should be compressed is quantified by the smooth conditional min-entropy, which we now define. For a state ρ_{AB} , the non-smooth conditional min-entropy is defined as

$$H_{\min}(A|B)_\rho := \max_{\sigma_B} \sup \{ \lambda \in \mathbb{R} : 2^{-\lambda} \mathbb{1}_A \otimes \sigma_B \geq \rho_{AB} \},$$

in terms of which the smooth min entropy is given by

$$H_{\min}^\epsilon(A|B)_\rho := \max_{\bar{\rho}_{AB}} H_{\min}(A|B)_{\bar{\rho}}.$$

The maximization over $\bar{\rho}$ is over a set of states that are close to ρ_{AB} according to some distance measure (see, for example, [34] for a discussion).

The significance for privacy amplification can be seen as follows. In [3], it is shown that if f is chosen randomly from a set of two-universal hash functions, and applied

1. Entangled quantum states used in the protocol are generated by a device Bob holds (which is separate and kept isolated from his measurement devices) and then shared over an insecure quantum channel with Alice's first device. Bob feeds his half of each state to his first measurement device. Once the states are received, the quantum channel is closed.
2. Alice and Bob each pick a random input A_i and B_i to their first device, ensuring they receive an output bit (X_i and Y_i respectively) before making the next input (so that the i -th output cannot depend on future inputs). They repeat this M times.
3. Bob publicly announces his measurement choices, and Alice checks that for a sufficient number of suitable input combinations for the protocol. If not, Alice aborts.
4. (*Sifting.*) Some output pairs may be discarded according to some protocol.
5. (*Parameter estimation.*) Alice and Bob use their pre-shared key to randomly select some output pairs (they select only a small fraction, hence the amount of key required for this is small). For each of the selected pairs, Bob encrypts his output and sends it to Alice. Alice uses the communicated bits and her corresponding outputs to compute some test function, and aborts if it lies outside a desired range.
6. (*Error correction.*) Alice and Bob perform error correction using public discussion, in order to (with high probability) generate identical strings. Eve learns the error correction function Alice applies to her string.
7. Alice and Bob repeat Steps 1–6 for each of their m devices (ensuring the devices cannot communicate throughout)
8. (*Privacy amplification.*) Alice and Bob concatenate their m strings and publicly perform privacy amplification [22], producing a shorter shared string about which Eve has virtually no information. In this step, the size of their final string is chosen such that (with high probability) it will remain secure even if one of the raw strings or its error corrected version becomes known.

TABLE 2: **Structure of the protocol from the main text with modifications as in Countermeasure 3.** For this protocol Alice and Bob each have $m \geq 2$ measurement devices, and Bob has one device for creating states. They are all kept isolated from one another.

to the raw string X , as above, then for $|S| = 2^t$ and any $\varepsilon \geq 0$,

$$D(\rho_{SRE}, \tau_S \otimes \rho_{RE}) \leq \varepsilon + \frac{1}{2} 2^{-\frac{1}{2}(H_{\min}^\varepsilon(X|E) - t)}.$$

(An analogous statement can be made for Trevisan's extractor [33].) Thus, if Alice compresses her string to length $t = H_{\min}^\varepsilon(X|E) - \ell$, then the final state after applying the hash function has distance $\varepsilon + \frac{1}{2} 2^{-\ell/2}$ to a state about which Eve has no knowledge.

Turning to the QKD protocol in Table 1 of the main

text, in the case of hashing the privacy amplification procedure consists of Alice selecting t depending on the test function computed in the parameter estimation step. She then uses local randomness to choose a hash function to apply to her string, and announces this to Bob, who applies the same function to his string (since we have already performed error correction, this string should be identical to Alice's). The idea is that, if t is chosen appropriately, it is virtually impossible that the parameter estimation tests pass and the final state at the end of the protocol is not close to one for which Eve has no knowledge about the final string.

In the modified protocol in Table 2, we expect each pair of devices to contribute roughly the same amount of smooth min entropy to the concatenated string. Thus, since there are m devices, in order to tolerate the potential revelation of one of the error-corrected strings through an abort attack, Alice should choose t to be roughly $(m - 1)/m$ shorter than she would otherwise.

Appendix D: Memory attacks on multi-device QKD protocols

To illustrate further the generality of our attacks, we now turn to multi-device protocols, and show how to break iterated versions of two well known protocols.

Attacks on compositions of the BHK protocol

The Barrett-Hardy-Kent (BHK) protocol [6] requires Alice and Bob to share MN^2 pairs of systems (where M and N are both large with $M \ll N$), in such a way that no measurements on any subset can effectively signal to the others. In a device-independent scenario, we can think of these as black box devices supplied by Eve, containing states also supplied by Eve. Each device is isolated within its own sub-laboratory of Alice's and Bob's, so that Alice and Bob have MN^2 secure sub-laboratories each. The devices accept integer inputs in the range $\{0, \dots, N - 1\}$ and produce integer outputs in the range $\{0, 1\}$. Alice and Bob choose random independent inputs, which they make public after obtaining all the outputs. They also publicly compare all their outputs except for those corresponding to one pair randomly chosen from among those in which the inputs differ by ± 1 or 0 modulo N . If the publicly declared outputs agree with quantum statistics for specified measurement basis choices (corresponding to the inputs) on a singlet state, then they accept the protocol as secure, and take the final undeclared outputs (which are almost certainly anticorrelated) to define their shared secret bit.

The BHK protocol produces (with high probability) precisely one secret bit: evidently, it is extremely inefficient in terms of the number of devices required. It also requires essentially noise-free channels and error-free measurements. Despite these impracticalities it il-

illustrates our theoretical point well. Suppose that Alice and Bob successfully complete a run of the BHK protocol and then (unauthorised by BHK) decide to use the same $2MN^2$ devices to generate a second secret bit, and ask Eve to supply a second batch of states to allow them to do this.

Eve — aware in advance that the devices may be reused — can design them to function as follows. In the first run of the protocol, she supplies a singlet pair to each pair of devices and the devices function honestly, carrying out the appropriate quantum measurements on their singlets and reporting the outcomes as their outputs. However, they also store in memory their inputs and outputs. In the second run, Eve supplies a fresh batch of singlet pairs. However, she also supplies a hidden classical signal identifying the particular pair of devices that generated the first secret bit. (This signal need go to just one of this pair of devices, and no others.) On the second run, the identified device produces as output the same output that it produced on the first run (i.e. the secret bit generated, up to a sign convention known to Eve). All other devices function honestly on the second run.

With probability $\frac{MN^2-1}{MN^2}$, the output from the cheating device on the second run will be made public, thus revealing the first secret bit to Eve. Moreover, with probability $1 - \frac{3}{2N} + O(N^{-2})$, this cheating will not be detected by Alice and Bob's tests, so that Eve learns the first secret bit without her cheating even being noticed.

There are defences against this specific attack. First, the BHK protocol [6] can be modified so that only outputs corresponding to inputs differing by ± 1 or 0 are publicly shared.⁵ While this causes Eve to wait many rounds for the secret bit to be leaked, and increases the risk her cheating will be detected, it leaves the iterated protocol insecure. Second, Alice and Bob could securely destroy or isolate the devices producing the secret key bit outputs, and reuse all their other devices in a second implementation. Since only the devices generating the secret key bit have information about it, this prevents it from being later leaked. While effective, this last defence really reflects the inefficiency of the BHK protocol: to illustrate this, we turn next to a more efficient multi-device protocol.

Attacks on compositions of the HR protocol

Hänggi and Renner (HR) [11] consider a multi-device QKD protocol related to the Ekert [2] protocol, in which Alice and Bob randomly and independently choose one of

two or three inputs respectively for each of their devices. If the devices are functioning honestly, these correspond to measurements of a shared singlet in the bases U_0, U_1 (Alice) and V_0, V_1, V_2 (Bob), defined by the following vectors and their orthogonal complements

$$\begin{aligned} U_1 &\leftrightarrow |0\rangle, \\ V_0 &\leftrightarrow \cos(\pi/8)|0\rangle + \sin(\pi/8)|1\rangle, \\ U_0, V_2 &\leftrightarrow \cos(\pi/4)|0\rangle + \sin(\pi/4)|1\rangle, \\ V_1 &\leftrightarrow \cos(3\pi/8)|0\rangle + \sin(3\pi/8)|1\rangle. \end{aligned}$$

The raw key on any given run is defined by the $\approx 1/6$ of the cases in which U_0 and V_2 are chosen. Information reconciliation and privacy amplification proceed according to protocols of the type described in the main text (in which the functions used are released publicly).

Evidently, our attacks apply here too if (unauthorised by HR) the devices are reused to generate further secret keys. Eve can identify the devices that generate the raw key on day 1, and request them to release their key as cheating outputs on later days, gradually enough that the cheating will be lost in the noise. Since the information reconciliation and privacy amplification functions were made public by Alice, she can then obtain the secret key. Even if she is unable to communicate directly with the devices for a long time (because they were pre-installed with a very large reservoir of singlets), she can program all devices to gradually release their day 1 outputs over subsequent days, and so can still deduce the raw and secret keys.

Alice and Bob could counter these attacks by securely destroying or isolating all the devices that generated raw key on day 1 — but this costs them $1/6$ of their devices, and they have to apply this strategy each time they generate a key, leaving $(5/6)^N$ of the devices after N runs, and leaving them able to generate shorter and shorter keys. As the length of secure key generated scales by $(5/6)^N$ (or worse, allowing for fluctuations due to noise) on each run, the total secret key generated is bounded by $\approx 6M$, where M is the secret key length generated on day 1.

Note that, as in the case of the iterated BHK protocol, all devices that generate secret key become toxic and cannot be reused. While the relative efficiency of the HR protocol ensures a (much) faster secret key rate, it also requires an equally fast device depletion rate. This example shows that our attacks pose a generic problem for device-independent QKD protocols of the types considered to date.

Appendix E: Device-independent randomness expansion protocols: attacks and defences

Device-independent quantum randomness expansion (DVI QRE) protocols were introduced by two of us [13, 15], developed further by [14, 35–37], and there now exist schemes with unconditional security proofs [36]. The

⁵ As originally presented, the BHK protocol requires public exchange of all outputs except those defining the secret key bit. This is unnecessary, and makes iterated implementations much more vulnerable to the attacks discussed here.

cryptographic scenario here is slightly different from that of key distribution in that there is only one honest party, Alice.

Alice’s aim is to expand an initial secret random string to a longer one that is guaranteed secret from an eavesdropper, Eve, even if the quantum devices and states used are supplied by Eve. The essential idea is that seed randomness can be used to carry out nonlocality tests on the devices and states, within one or more secure laboratories, in a way that guarantees (with numerical bounds) that the outcomes generate a partially secret and random string. Privacy amplification can then be used to generate an essentially fully secret random string, which (provided the tests are passed) is significantly longer than the initial seed.

There are already known pitfalls in designing such protocols. For example, although one might think that carrying out a protocol in a single secure laboratory guarantees that the initially secure seed string remains secure, and so guarantees randomness expansion if any new secret random data is generated, this is not the case [15]. Eve’s devices may be programmed to produce outputs depending on the random seed in such a way that the length of the final secret random string depends on the initial seed. Protocols with this vulnerability are not compositely secure. (To see this can be a practical problem, note that Eve may infer the length of the generated secret random string from its use.)

A corollary of our results is that, if one wants to reuse the devices to generate further randomness, it is crucial to carry out DVI QRE protocols with devices permanently held within a *single* secure laboratory, avoiding any public communication of device output data at any stage. It is crucial too that the devices themselves are securely isolated from classical communications and computations within the laboratory, to prevent them from learning details of the reconciliation and privacy amplification.

Even under these stringent conditions, our attacks still apply in principle. For example, consider a noise-tolerant protocol that produces a secret random output string of variable length, depending on the values of test functions of the device outputs (the analogue of QKD parameter estimation for QRE) that measure how far the device outputs deviate from ideal honest outputs. This might seem natural for any single run, since – if the devices are never reused – the length of the provably secret random string that can be generated does indeed depend on the value of a suitable test function. However, iterating such

a protocol allows the devices to leak information about (at least) their raw outputs on the first run by generating artificial noise in later rounds, with the level of extra noise chosen to depend suitably on the output values. Such noise statistically affects the length of the output random strings on later rounds.

In this way, suitably programmed devices could ultimately allow Eve to infer all the raw outputs from the first round, given observation of the key string lengths created in later rounds. This makes the round one QRE insecure, since given the raw outputs for round one, and knowing the protocol, Eve knows all information about the output random string for round one, except that determined by the secret random seed.

One defence against this would be to fix a length L for the random string generated corresponding to a maximum acceptable noise level, and then to employ the Procrustean tactic of always reducing the string generated to length L , regardless of the measured noise level.

Even then, though, unless some restriction is placed on the number of uses, the abort attack on QKD protocols described in the main text also applies here. The devices have the power to cause the protocol to abort on any round of their choice, and so – if she is willing to wait long enough – Eve can program them to communicate any or all information about their round 1 raw outputs by choosing the round on which they cause an abort.

We also described in the main text a moderately costly but apparently effective defence against abort attacks on QKD protocols, in which Alice and Bob each have several isolated devices that independently generate raw sub-keys, which are concatenated and privacy amplified so that exposing a single sub-key does not significantly compromise the final secret key. This defence appears equally effective against abort attacks on device-independent quantum randomness expansion protocols. Since quantum randomness expansion generally involves only a single party, these protocols are not vulnerable to the impostor attacks described in the main text. It thus appears that it may be possible in principle to completely defend them against memory attacks, albeit at some cost.

It is also worth noting that there are many scenarios in which one only needs short-lived randomness, for example, in many gambling applications, bets are often placed about random data that are later made public. In such scenarios, once such random data have been revealed, the devices could be reused without our attacks presenting any problem.